**结算平台各系统对接配置中心技术设计文档**

版本 **1.0**

北京弘合柏基金融信息服务有限责任公司

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本 | 说明 | 作者 |
| 2019-09-18 | 1.0 | 创建文档 | 陈丽霞 |
|  |  |  |  |

# 1 需求背景

* 运维：
* 1. 生产上线替换90%操作(除三方密钥文件包)可以由配置中心解决相应的问题；类似支付三方文件方案待定；
* 2.相应的解决方案架构组都准备完毕，针对各个功能本周都是做了一一验证，随时可以配合各组进行改造；
* 3.根据搜集各组研发系统情况，各组改造的时间点需要由各研发组根据自己的实际情况进行待排期，此项改造与k8s迁移没有必然关系；
* 4.与成都做了沟通确认，在上线替换操作上，目前的进度情况基本一致; 这方面两边是基本同步的；
* 5.待各组改造完成后，配置中心的生产运维管理权限会依然由运维人员进行维护，从安全考虑暂不下放给研发。

# 2 交互流程

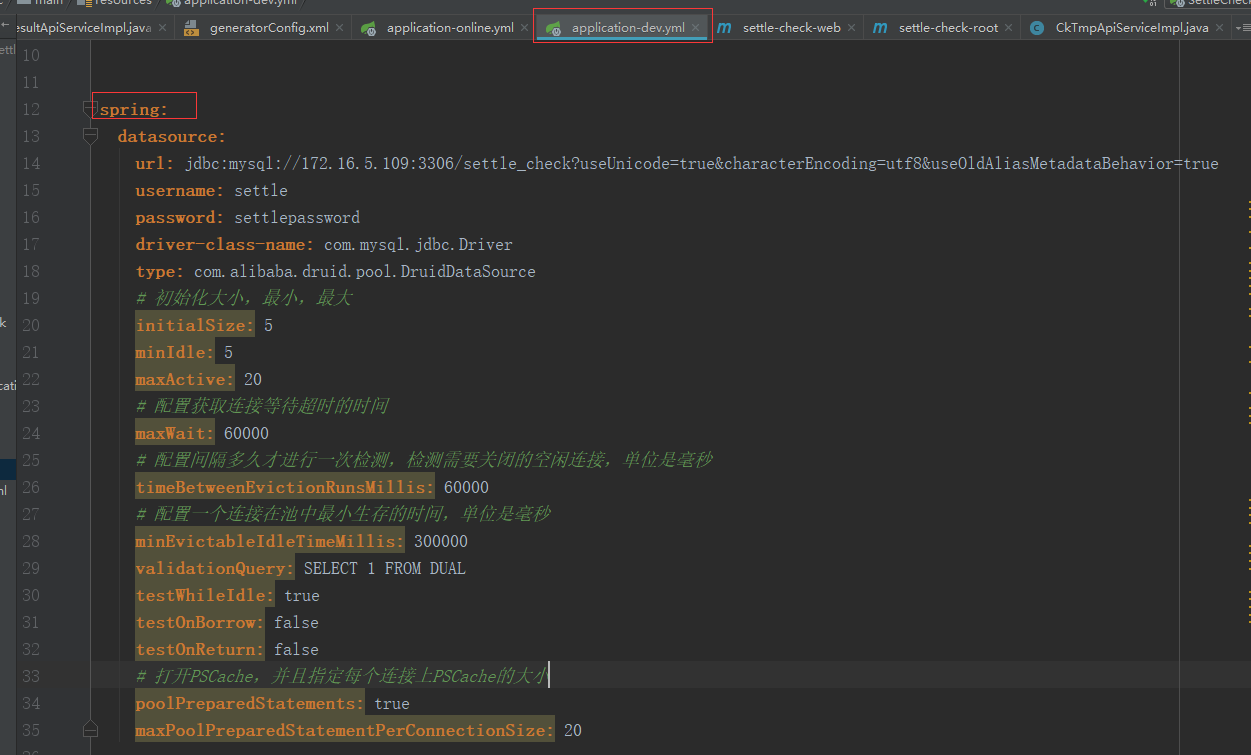
# 3 功能设计

### 3.1 模块设计

Springboot系统对接配置中心实现数据库配置信息中心化，数据库连接池druid。

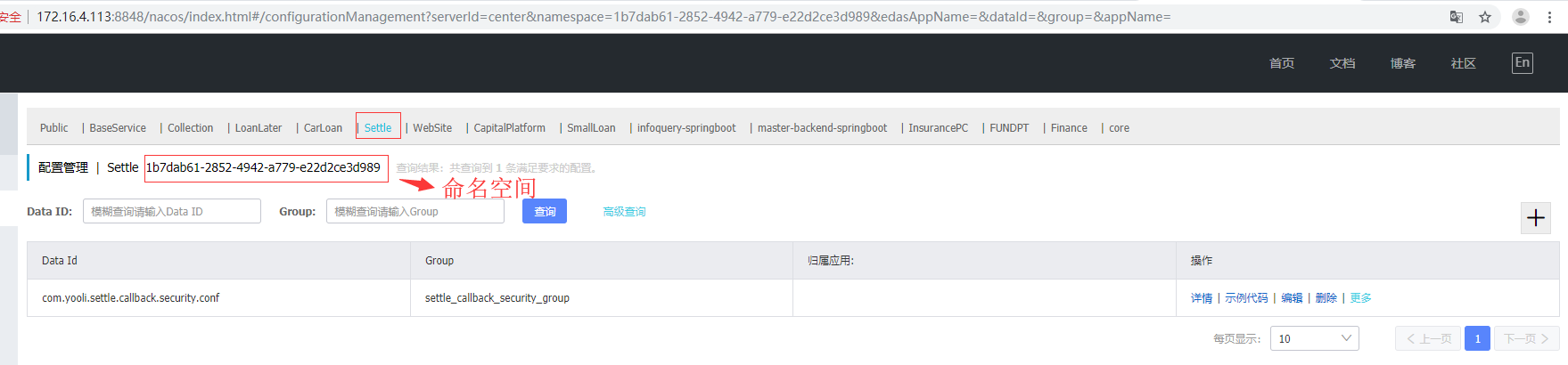
**一、现有数据库连接方式**

* application-dev\deta\online.yml或者properties文件，数据库连接池druid。

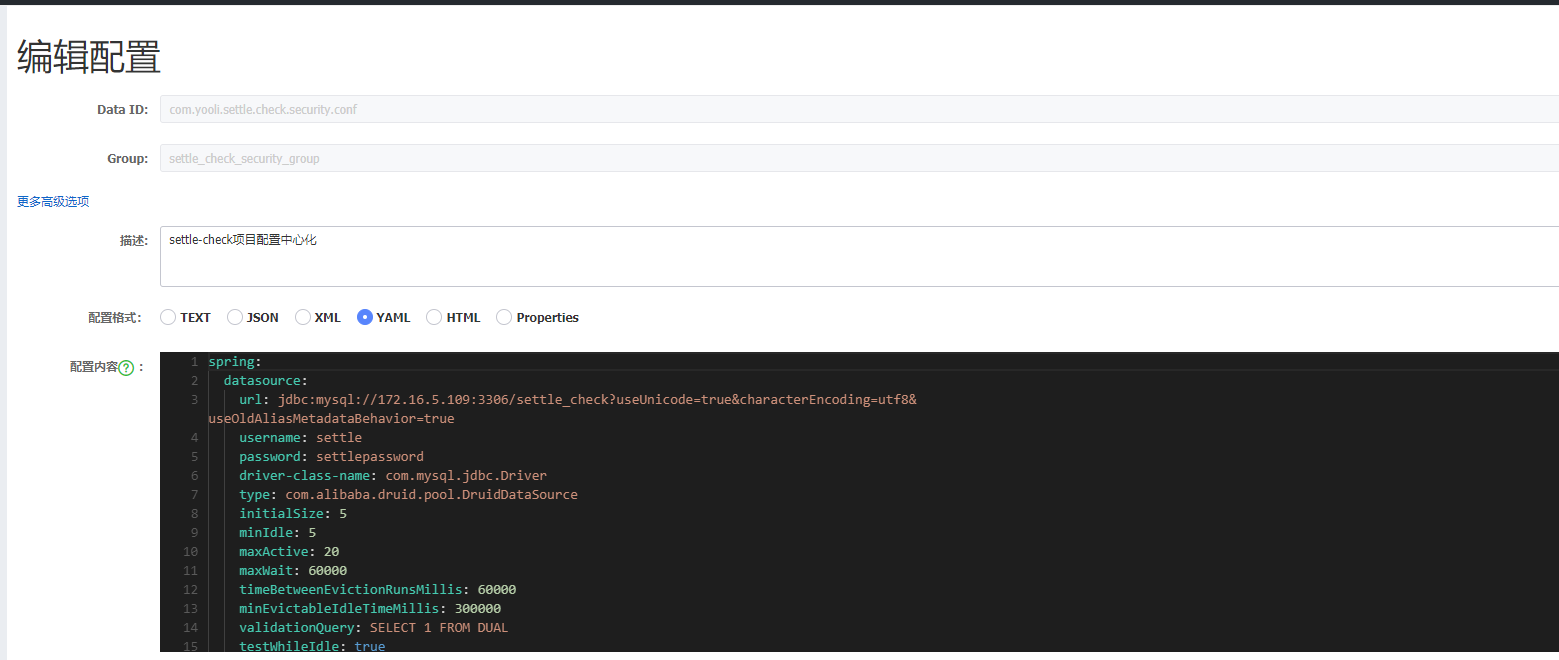


**二、数据库配置中心化**

* nacos测试：<http://172.16.4.113:8848/nacos/index.html，各个项目组nacos对应的命名空间由架构组分配，每一个命名空间可以有多个groupId和dataId,groupId和dataId一对一关系。>



* nacos配置中心groupId和dataId已申请完毕，开发及生产环境使用settle\_check\_datasource\_security\_conf和settle\_check\_security\_group即可。



* nacos对应jar包

<dependency>

<groupId>com.alibaba.nacos</groupId>

<artifactId>nacos-client</artifactId>

<version>1.1.3</version>

<exclusions>

<exclusion>

<artifactId>fastjson</artifactId>

<groupId>com.alibaba</groupId>

</exclusion>

</exclusions>

</dependency>

<dependency>

<groupId>com.alibaba.nacos</groupId>

<artifactId>nacos-spring-context</artifactId>

<version>0.3.3</version>

</dependency>

* nacos对应参数配置引入方式（区分dev\beta\online环境）

yml方式引入，列如：application-dev\deta\online.yml或者properties增加

nacos:

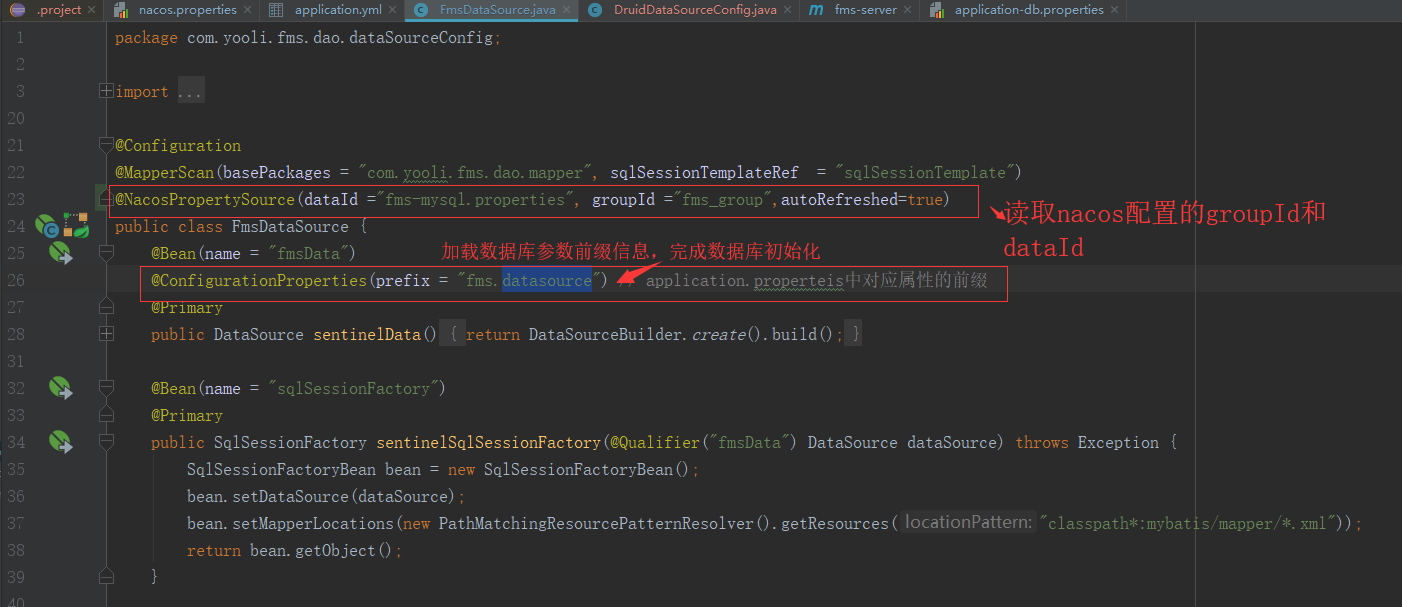
config:

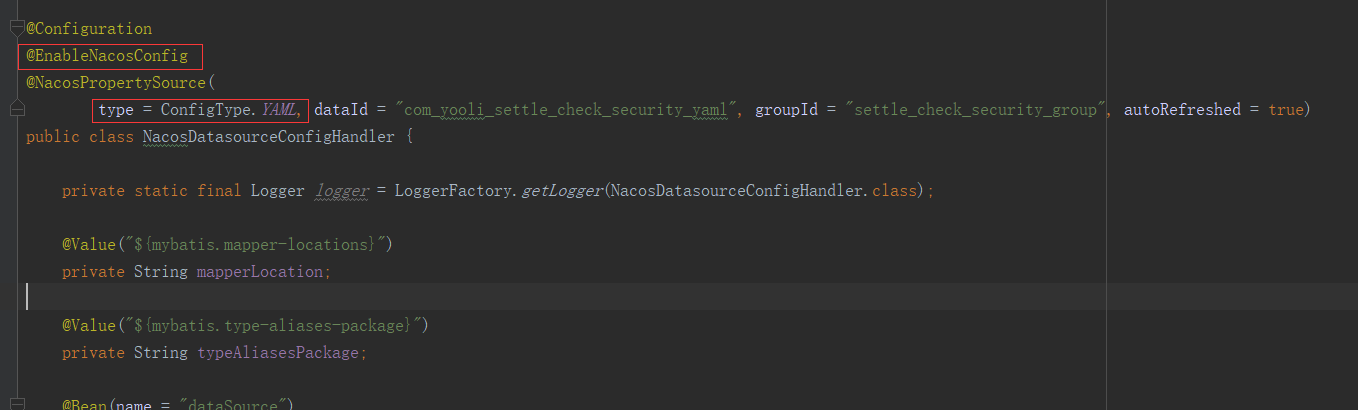
server-addr: 172.16.4.113:8848

namespace: 1b7dab61-2852-4942-a779-e22d2ce3d989

* 数据库连接池druid初始化

结合目前route项目的数据库连接池初始化类，区别在于数据库信息需要从nacos配置中心获取，route项目参考代码：com.yooli.settle.route.config.DataSourceConfig，拿取nacos配置中心数据库文件见下图。





@EnableNacosConfig

@NacosPropertySource(

type = ConfigType.YAML, dataId = "", groupId = "", autoRefreshed = true)

@ConfigurationProperties(prefix = "")

* redis初始化

1. xml方式：Xml配置文件增加引入nacos，拿取nacos配置中心参数见下图。



<!--nacos配置，这里是使用配置文件的方式,这只是其中的一种方式-->

<!--开启注解-->

<nacos:annotation-driven></nacos:annotation-driven>

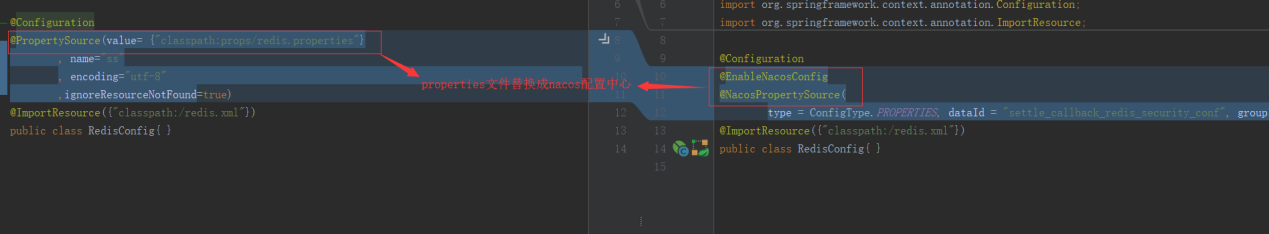
<!--指定nacos配置地址-->

<nacos:global-properties server-addr="${nacos.serverAddr}" namespace="${nacos.namespace}"/>

<!--指定dataId,group-id, 是否是自动刷新-->

<nacos:property-source data-id="${nacos.conf.redis.dataId}" group-id="${nacos.conf.redis.group}" auto-refreshed="true"/>

B.注解方式：



@EnableNacosConfig

@NacosPropertySource(

type = ConfigType.YAML, dataId = "", groupId = "", autoRefreshed = true)

**三、注意点**

* 上线前需和架构组提前沟通开通生产nacos的groupId和dataId，并相应的配置所对应数据库连接参数；
* Application-dev\beta\online.yml或者properties剔除数据库配置信息，移交到nacos配置中心。
* Application-dev\beta\online.yml或者properties剔除redis配置信息，移交到nacos配置中心。

### 3.2配置信息

application-dev\deta\online.yml或者properties文件

### **3.3影响范围**（关联系统影响和业务影响）

结算平台 settle\_check、settle\_callback、settle\_route

# 4 交互时序图

# 5 接口设计

# 5 存储设计

## 5.1 数据库

5.2 数据迁移

# 6 运维支持

## 6.1 服务器

<是否需要申请服务器，对服务器的要求说明>

## 6.2 网络

## 6.3 权限

<需要申请的对文件的目录的权限说明等>

## 6.4 域名和证书

# 7 监控设计

<对于功能运行新能参数及可用性的监控设计>

# 8 系统物理部署

<图文描述系统在物理节点上详细的部署结构，部署拓扑图，新增必选>

# 9 兼容性及异常处理

# 10 安全性

# 11 其他

<其他需要说明的问题汇总>